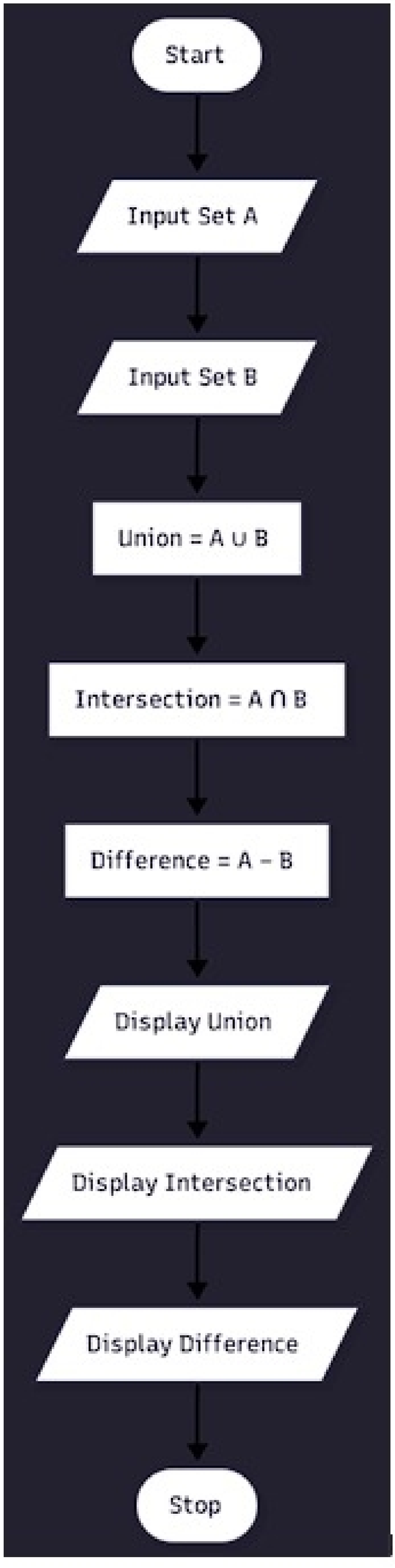
Algorithm: Set Operations (Union, Intersection, Difference)

1. **Start**
2. **Input Set A**
3. **Input Set B**
4. **Compute Union**
    - Union = A ∪ B
5. **Compute Intersection**
    - Intersection = A ∩ B
6. **Compute Difference**
    - Difference = A − B
7. **Display Union**
8. **Display Intersection**
9. **Display Difference**
10. **Stop**


Start
Input Set A
Input Set B
Union = A ∪ B
Intersection = A ∩ B
Difference = A − B
Display Union
Display Intersection
Display Difference
Stop

4.1.1. Set Operations

18:05

Write a Python program to perform union, intersection and difference operations on $Set\ A$ and $Set\ B$.

Input Format:

- First Line prompts "Set A: " followed by space-separated list of integers for $Set\ A$.
- The second input prompts "Set B: " followed by space-separated list of integers for $Set\ B$.

Output Format:

- The first line prints "Union: " followed by the union of $Set\ A$ and $Set\ B$.
- The second line prints "Intersection: " followed by the intersection of $Set\ A$ and $Set\ B$.
- The third line prints "Difference: " followed by the difference of $Set\ A$ and $Set\ B$.

Note:

- If there is no intersection between the two sets, the program prints an empty set, which appears as "set()" in the output.
- Please refer to the visible test cases for better understanding.

Sample Test Cases +

Explorer

setoperat...

Submit

Debugger

```
# Type Content here...
set_a = set (map(int, input("Set A: ").split()))
set_b = set (map(int, input("Set B: ").split()))

union_set = set_a | set_b
intersection_set = set_a & set_b
difference_set = set_a - set_b

print("Union:", union_set)
print("Intersection:", intersection_set)
print("Difference:", difference_set)

```

Terminal Test cases

‹ Prev Reset Submit Next ›